Edizione italiana
sinclair
ZX SPECTRUM
Microdrive and Interface 1 manual

# SINCLAIR
# INTERFACCIA ZX 1
# MICRODRIVE ZX

Cambridge Comunication Limited

traduzione di

Giacomo Bortone

a cura di

Rita Bonelli

Edizioni JCE

Via dei Lavoratori, 124

CINISELLO BALSAMO (MI)

---

INTRODUZIONE

DI CHE COSA SI TRATTA

---

L' INTERFACCIA ZX 1...
svolge tre diverse funzioni, come controllore per i Microdrive, come rete di lavoro locale e come interfaccia RS232. Collegata allo Spectrum permette di controllare fino a otto Microdrive, di scambiare dati con altri calcolatori e di pilotare una vasta gamma di stampanti.

LO ZX MICRODRIVE...
mette a disposizione una grande quantita' di memoria di massa a funzionamento molto piu' veloce delle tradizionali cassette. I Microdrive funzionano a cartucce di nastro mobili, ognuna delle quali puo' contenere fino a 100K byte.

QUESTO MANUALE...
spiega come ottenere dal BASIC e dal sistema operativo dello Spectrum alcune funzioni tipiche di sistemi piu' costosi; come aprire dei canali di comunicazione per trasmettere e ricevere dati e gestire i file. Le nuove funzioni aumentano enormemente le capacita' della macchina in tutte le operazioni in cui:
- vengono memorizzati o caricati dati o programmi,
- dati e programmi sono inviati verso delle periferiche,
- sono stabilite delle linee di comunicazione con altri Spectrum o con altri calcolatori.

L'uso combinato del BASIC come linguaggio di programmazione e come attivatore di funzioni particolari del sistema operativo, conferisce allo Spectrum esteso una potenza, una flessibilita' e anche una facilita' d'uso tipica di computer molto piu' grandi.

Vi consigliamo di leggere il capitolo sui Microdrive anche se non ne avete comperato uno, troverete utili spiegazioni sui concetti di "canale" e "flusso".

Questo manuale deve essere letto soltanto da chi ha gia' familiarita' con lo ZX Spectrum e abbia letto il manuale del calcolatore.

INDICE

---

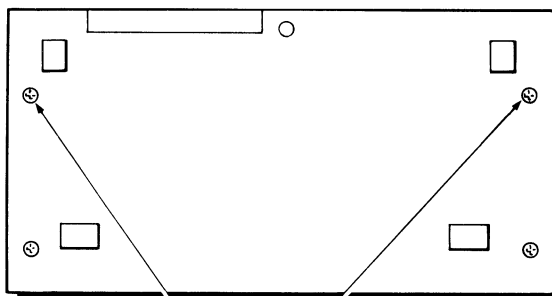
# 1. INSTALLAZIONE DELL'INTERFACCIA ZX 1

---

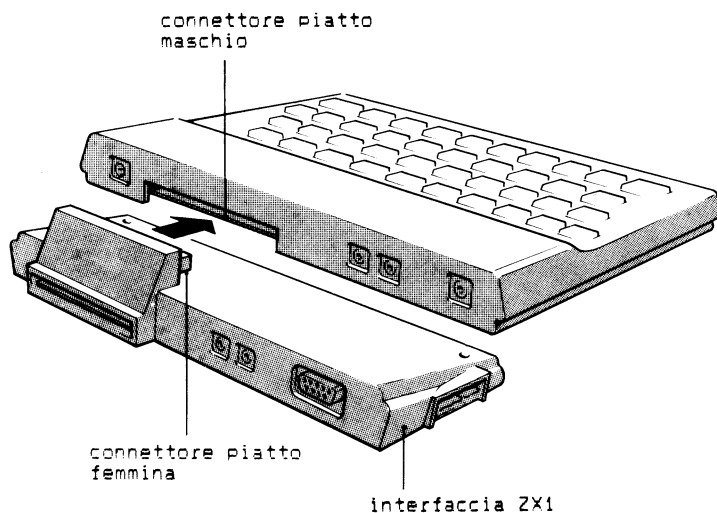
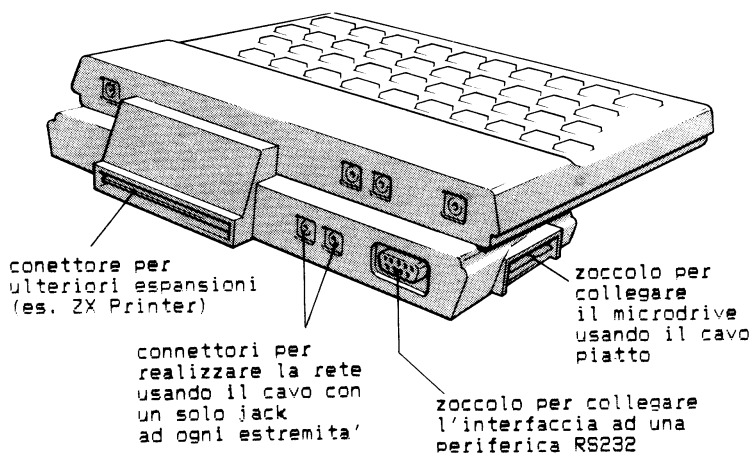
La scatola dell' interfaccia contiene:

- questo libro di spiegazioni,
- l'interfaccia (fornita con due viti lunghe),
- un cavo piatto lungo circa 8 cm. (per il collegamento con il Microdrive),
- un cavo con uno spinotto jack ad ogni estremita' (per stabilire un collegamento con un altro calcolatore equipaggiato di interfaccia).

Per collegare l'interfaccia allo Spectrum occorre:

1. Spegnere il calcolatore.
2. Scollegarlo da tutte le periferiche.
3. Togliere con un cacciavite a croce le due viti sotto il computer come mostrato nello schema, esse non occorreranno piu' a meno che non si smonti l'interfaccia.

4. Inserire saldamente il connettore piatto dell'interfaccia nel connettore posto sul retro dello Spectrum.

5. Avvitare l'interfaccia sotto il calcolatore usando le viti lunghe.

Ora e' possibile ricollegare tutte le periferiche allo zoccolo posto sul retro dell' interfaccia e accendere il calcolatore.

L'interfaccia mette a disposizione quattro nuovi connettori; servono per:

Attraverso di essi si puo':

- collegare un Microdrive...............(vedi cap.2)
- realizzare una rete...................(vedi cap.7)
- collegarsi ad una periferica RS232....(vedi cap.8)

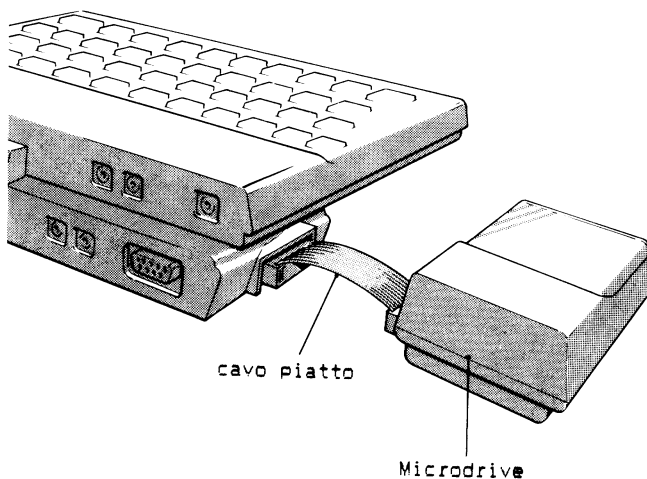
---

## 2. INSTALLAZIONE DEL MICRODRIVE

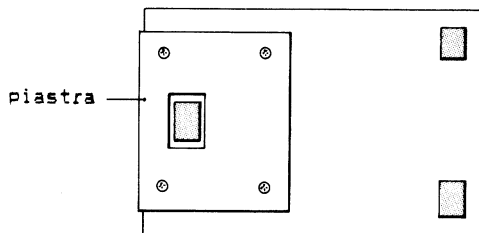
---

La scatola del Microdrive contiene:

- il Microdrive,
- un connettore per collegarlo ad un altro Microdrive,
- un cartridge dimostrativo preregistrato.

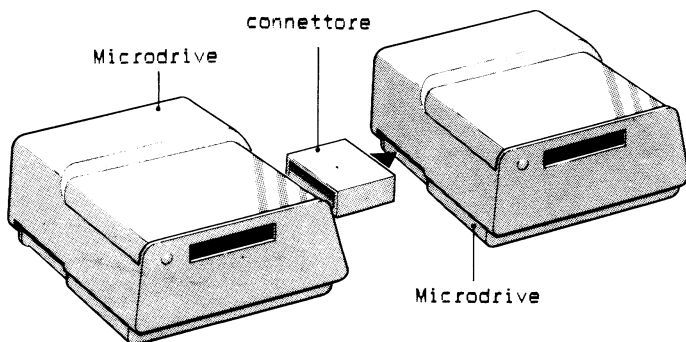
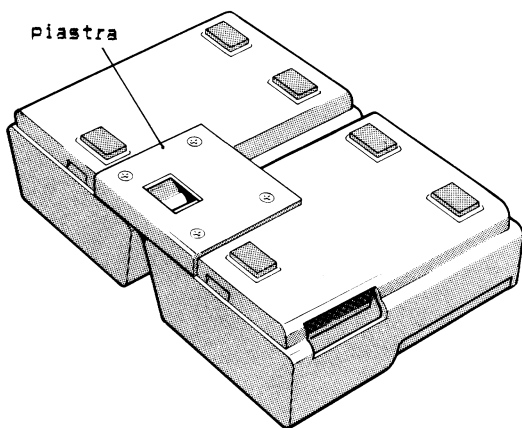
Per collegare il Microdrive all'interfaccia inserire una estremita' del cavo piatto fornito nello zoccolo posto sul lato di questa e l'altra estremita' del cavo nello zoccolo posto al lato destro del Microdrive. Assicuratevi che le connessioni siano ben stabili.

Per collegare piu' Microdrive insieme occorre:

1. Scollegare il primo Microdrive dal calcolatore.

2. Rovesciare i due Microdrive e metterli uno di fianco all'altro. Si notera' subito una piastra avvitata sotto ogni Microdrive.

3. Svitare e togliere entrambe le piastre.

4. Riportare i Microdrive alla loro posizione normale.

5. Inserire stabilmente il connettore di uno dei due nel lato di uno dei Microdrive (come mostrato nella figura).

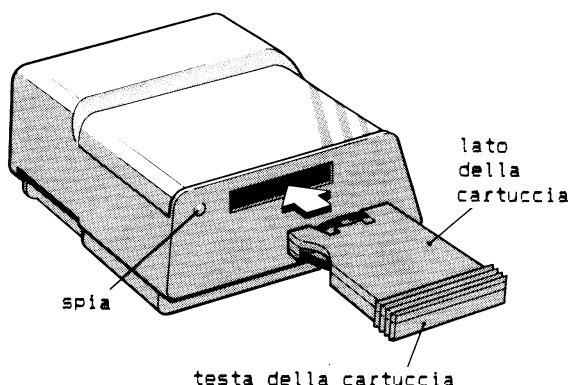
6. Infilare l'altro Microdrive sul connettore appena infilato sul primo in modo tale che i due Microdrive risultino affiancati.

7. Rovesciare con delicatezza i due Microdrive e avvitare una delle placche in modo tale che li blocchi uno attaccato all'altro. Avanzano una placca e un connettore che possono essere utilizzati per collegare un eventuale terzo Microdrive.

8. Riportare i Microdrive in posizione normale e ricollegare quello sul lato destro al calcolatore.

Si noti che il Microdrive piu' vicino al computer e' sempre il Microdrive 1 e che quello di fianco e' il Microdrive 2 e cosi' via.

Si noti anche che davanti ad ogni Microdrive c'e' una spia luminosa che si accende durante il funzionamento.

LE CARTUCCE

Ogni cartuccia viene fornita in una scatola protettiva dove deve essere sempre riposta quando non e' inserita nel Microdrive.

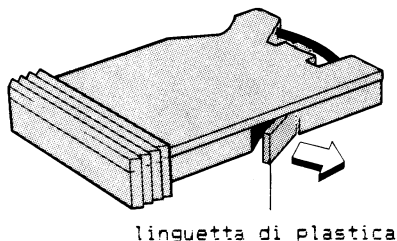
Quando la cartuccia viene estratta dalla sua scatola occorre fare attenzione a non toccare il nastro con le dita. Sulla cartuccia sono applicate due etichette, una in testa e una sulla parte superiore.

Per caricare il Microdrive infilare FINO IN FONDO la cartuccia, facendo attenzione che l'etichetta sia rivolta verso l'alto. La testa della cartuccia rimarra' sporgente di circa mezzo centimetro.

Quando avete finito di usare una cartuccia estraetela delicatamente dal Microdrive e rimettetela nella sua scatola protettiva.

SI FACCIA MOLTA ATTENZIONE A NON ESTRARRE MAI LA CARTUCCIA DAL MICRODRIVE QUANDO LA LUCE E' ACCESA E A NON ACCENDERE O SPEGNERE MAI UN MICRODRIVE (QUINDI LO SPECTRUM) CON UNA CARTUCCIA INSERITA.

## COME PROTEGGERE I PROGRAMMI E I DATI MEMORIZZATI IN UNA CARTUCCIA

Per proteggere i programmi e i dati memorizzati in una cartuccia, ovvero per evitare che possano essere cancellati o modificati, e' sufficiente spezzare la linguetta di plastica posta sul lato destro della cartuccia stessa, come mostrato nella figura.

linguetta di plastica

Questo sistema di protezione funziona come quello delle cassette normali. Per poter riscrivere su una cartuccia protetta occorre chiudere lo spazio lasciato dalla linguetta con un pezzo di nastro adesivo.

## VITA MEDIA DI UNA CARTUCCIA

Le cartucce dei Microdrive non sono eterne, prima o poi si consumano e devono essere sostituite. Quando una cassetta comincia a diventare vecchia ed e' ora di sostituirla, il calcolatore avra' sempre piu' difficolta' a trovare i dati memorizzati e impieghera' sempre piu' tempo per caricarli. Si consiglia di tenere sempre una copia dei programmi e dei dati importanti su un'altra cartuccia o su una cassetta, come si fa normalmente lavorando con i floppy disk.

---

## 3. PRIME OPERAZIONI CON I MICRODRIVE

---

### CARICAMENTO AUTOMATICO

Dopo aver installato l'interfaccia e il Microdrive e' possibile esaminare la cartuccia dimostrativa. Inseritela nel Microdrive (o nel Microdrive 1 se ne avete diversi) e scrivete:

NEW

seguito da

RUN (e ENTER)

La luce del Microdrive si accende e viene automaticamente caricato e lanciato il primo programma della cartuccia; fatelo girare prima di proseguire nella lettura.

### IL CATALOGO

Per sapere quali programmi sono memorizzati nella cartuccia dimostrativa occorre dare il comando di CATalogo:

CAT 1 (1 identifica il Microdrive numero 1)

Dopo circa 7 secondi sullo schermo apparira':

- il nome della cartuccia
- una lista in ordine alfabetico dei nomi dei file memorizzati nella cartuccia
- la quantita' di memoria non utilizzata nella cartuccia espressa in kilobyte.

## CARICAMENTO DEI PROGRAMMI

Potete scegliere e caricare uno qualunque dei programmi del catalogo. Scegliete un programma e scrivete:

```
LOAD *"m";1;"nome"
```

L'asterisco indica che l'operazione di LOAD deve essere eseguita attraverso l'interfaccia e non dal registratore a cassette.

"m";1; identifica il Microdrive numero 1

"nome" e' il nome del file che avete scelto, come per le operazioni su cassetta.

Dopo un attimo lo schermo mostra il messaggio OK ma non il nome del programma come avveniva con le cassette. Ora potete far girare il programma con RUN o listarlo con LIST.

-----------------------------

## 4. I PROGRAMMI E IL MICRODRIVE

-----------------------------

### MEMORIZZARE, VERIFICARE, CARICARE E MISCELARE PROGRAMMI

Nel manuale di istruzioni dello ZX Spectrum e' stato spiegato come memorizzare i programmi usando il registratore a cassette. Memorizzare un programma su una cartuccia e' altrettanto facile. Ecco un programma di nome "quadrati" che assomiglia molto ad uno che dovreste gia' conoscere:

```
10 REM quadrati
20 FOR n=1 TO 10
30 PRINT n,n*n
40 NEXT n
```

Per memorizzare questo su una cassetta avreste dovuto scrivere:

```
SAVE "quadrati"
```

Per memorizzarlo sulla cartuccia del Microdrive 1 occorre invece scrivere:

```
SAVE *"m";1;"quadrati"
```

Dopo circa 6 secondi il bordo dello schermo lampeggia e il programma viene registrato. Provate e verificate.
Come nel caso delle cassette i nomi dei file registrati su cartuccia possono essere al massimo di dieci caratteri. Come con le cassette e' possibile verificare se il programma e' stato memorizzato correttamente con il comando:

```
VERIFY *"m";1;"quadrati"
```

Se tutto e' avvenuto correttamente lo schermo mostrera' il messaggio "OK".
Per caricare il programma "quadrati" occorre scrivere:

```
LOAD *"m";1;"quadrati"
```

Ricordatevi di dare prima il NEW.

Come gia' visto con le cassette e' possibile fare in modo che il programma parta automaticamente appena

caricato. Provate ad eseguire:

```
SAVE *"m";1;"quadrati2" LINE 10
```

poi:

```
NEW
```

seguito da:

```
LOAD *"m";1;"quadrati2"
```

I programmi possono essere miscelati anche usando i Microdrive, come con le cassette. Scrivete:

```
NEW
```

poi battete:

```
100 REM altri quadrati
110 FOR n=11 TO 20
120 PRINT n,n*n
130 NEXT n
```

e scrivete:

```
MERGE *"m";1;"quadrati"
```

Il programma "quadrati" sara' aggiunto al listato che avete appena introdotto.
In pratica con i Microdrive si possono usare tutte le istruzioni Basic che si usano con le cassette e che sono spiegate nel capitolo 29 del manuale di programmazione in italiano.

## CANCELLAZIONE DI PROGRAMMI

Se avete memorizzato il programma "quadrati" e lo volete cancellare scrivete:

```
ERASE "m";1;"quadrati"
```

(come gia' detto "m";1; indica che state usando il Microdrive 1)

Durante l'esecuzione del comando ERASE il bordo dello schermo lampeggia.
Se si commette l'errore di usare il comando BREAK durante la memorizzazione del programma, sulla cartuccia

rimane un file non chiuso che non puo' essere caricato e ritorna sempre il messaggio di "File not found". Il comando ERASE puo' cancellare anche un file non chiuso ma impiega circa 30 sec. perche' il calcolatore controlla la cartuccia diverse volte per cercare la fine (inesistente) del file.

FORMATTAZIONE E ETICHETTATURA DELLE CARTUCCE VERGINI

Una cartuccia vergine deve essere formattata prima di poter essere impiegata; per fare cio' occorre inserirla nel Microdrive e scrivere:

```
FORMAT "m";1;"nome"
```

("m";1; indica come sempre il Microdrive 1 e "nome" e' il nome che si vuole dare alla cartuccia. Il nome puo' essere lungo fino a 10 caratteri ).

Il calcolatore impiega circa 30 sec. per formattare una cartuccia. Durante questa operazione il bordo dello schermo lampeggia poi rimane stabile, poi lampeggia ancora e alla fine viene mostrato il messaggio OK. Durante l'operazione di formattazione il computer controlla tutto il nastro e fa in modo che le zone di nastro rovinate, non adatte alla memorizzazione, non vengano mai usate.

Di solito non c'e' bisogno di riformattare una cartuccia gia' formattata; si noti comunque che se si riformatta una cartuccia gia' usata, tutte le informazioni precedentemente memorizzate vengono cancellate. Se avete piu' Microdrive potete usare le cartucce dimostrative doppie per i vostri programmi, formattandole.

Dopo aver formattato una cartuccia provate a scrivere:

```
CAT 1
```

Dopo pochi sec. lo schermo televisivo mostra il nome della cartuccia e la sua capacita' di contenere dati. La capacita' e' leggermente variabile tra una cartuccia e l'altra ma non sara' mai inferiore ad 85 kilobyte; potrebbe arrivare oltre i 100k.

AUTO-RUN

Il primo programma della cassetta dimostrativa e' memorizzato con l'auto-run, quindi viene caricato e lanciato automaticamente quando viene scritto RUN subito dopo NEW.

Qualunque programma di uso frequente puo' essere memorizzato con questa opzione, in modo tale che possa venire caricato senza bisogno di ripetere la sequenza LOAD e RUN. Per fare cio' occorre che:

- il programma abbia nome "run";

- la cartuccia sia inserita nel Microdrive 1;

- il comando RUN sia dato subito dopo l'accensione o dopo un comando NEW.

Per memorizzare un programma in questo modo si deve scrivere:

```
SAVE *"m";1;"run" LINE numero
```

Il nome del file deve essere "run" scritto carattere per carattere e non attraverso il tasto RUN.
Numero deve essere il numero di linea dove si vuole che inizi l'esecuzione.

Ora scrivete:

```
NEW
```

seguito da:

```
RUN (e ENTER )
```

(Il comando RUN questa volta deve essere dato col tasto "R").
Un'ultima osservazione: il comando MERGE non funziona con i programmi memorizzati con l'opzione SAVE*...LINE. Questo e' stato fatto per permettere di proteggere tali programmi. Cercare di usare MERGE con un programma cosi' protetto da' "MERGE ERROR".

---

## 5. DATI CANALI E FLUSSI

---

Come sapete, un programma e' formato da una serie di istruzioni che viene eseguita quando date il RUN, e i dati sono un insieme di lettere, di numeri o di simboli su cui il programma svolge delle elaborazioni. Sono dati, per esempio, i numeri da 1 a 10 e i relativi quadrati.

I dati possono essere inviati e provenire da diverse parti di un sistema. Tali parti si chiamano CANALI. I canali verso i quali lo Spectrum puo' trasmettere sono:

- lo schermo televisivo,
- la stampante ZX Printer,
- un file sul Microdrive, a scopo di memorizzazione,
- un altro ZX Spectrum, se entrambi sono collegati alla stessa rete,
- l'interfaccia RS232 che a sua volta puo' essere collegata ad un modem o ad una stampante.

I canali dai quali i dati possono provenire sono:

- la tastiera,
- un file sul Microdrive,
- un altro ZX Spectrum, se entrambi sono collegati alla stessa rete,
- l'interfaccia RS232, collegata, per es., ad un modem o ad un terminale.

I percorsi che i dati seguono per arrivare e provenire da questi canali si chiamano FLUSSI. In un sistema Spectrum possono essere usati contemporaneamente al massimo 16 flussi, numerati da 0 a 15. Il numero di flusso e' sempre preceduto dal carattere #.

Quattro di questi flussi sono automaticamente collegati a dei canali:

| | |
|---|---|
| flusso #0 | stampano i dati nella parte bassa dello schermo e li leggono |
| flusso #1 | dalla tastiera |

flusso #2 scrive i dati nella parte superiore dello schermo ma non puo' leggere dati

flusso #3 scrive i dati alla stampante ZX Printer ma non puo' leggere dati.

Ogni istruzione di input o di output usa automaticamente uno di questi flussi. Il comando PRINT usa il flusso #2, e il comando LPRINT usa il flusso #3. Quindi il comando:

```
PRINT "Questo e' un calcolatore Spectrum"
```

Puo' essere considerato un' abbreviazione di:

```
PRINT #2; "Questo e' un calcolatore Spectrum"
```

Provate ad eseguire entrambe le linee e osservate che non c'e' alcuna differenza.

Anche se ogni istruzione di stampa e' gia' assegnata ad un flusso (PRINT ha il #2, LPRINT ha il #3, ecc.), e' possibile indirizzarla verso un qualunque altro flusso scrivendo # seguito dal numero desiderato. Provate:

```
LPRINT #2;"Questo e' un calcolatore Spectrum"
```

che produce una stampa sul video nonostante si sia usato il comando LPRINT.

I flussi dal #4 al #15 non sono predefiniti ma possono essere usati esattamente come gli altri. Prima di usarli naturalmente occorre indirizzarli verso un canale. Ognuno dei canali esistenti nel sistema Spectrum ha un nome, consistente in un carattere. I canali piu' usati sono:

"K" per la tastiera (Keyboard)

"S" per lo schermo (Screen)

"P" per la ZX Printer

(in seguito se ne vedranno altri).

Osservate che K, S e P sono canali pre-definiti. Con questi e' necessario usare la virgola come separatore in un comando OPEN #. Con altri canali e' invece possibile usare anche il punto e virgola.

Per assegnare un flusso a un canale si usa il comando OPEN# con questa sintassi:

```
OPEN #4,"S"
```

che assegna il flusso #4 al canale S. Come conseguenza il comando:

```
PRINT #4;"Questo e' un calcolatore Spectrum"
```

indirizza il messaggio verso il video.

Si faccia attenzione a non usare il comando OPEN con i flussi #0, #1 o #2 perche' cio' potrebbe provocare anomalie di funzionamento imprevedibili.

Si noti che il nome dei canali puo' essere indicato indifferentemente in lettere maiuscole e minuscole.

---------------------------

# 6. IL MICRODRIVE E I DATI

---------------------------

## COME APRIRE UN FILE DI DATI

Un file e' un qualunque insieme di informazioni memorizzato su una cartuccia, ed ha un nome per poterlo distinguere da tutti gli altri file. Per poter usare un file occorre che prima venga aperto, che sia cioe' definito come nuovo canale del sistema e assegnato ad un flusso. Deve anche essere stabilito su quale supporto fisico, per es. su quale Microdrive, debba risiedere. Tutte queste operazioni vengono eseguite dal comando:

```
OPEN #4:"m";1;"numeri"
```

Si noti che:

- il numero di canale (dopo il #) puo' essere qualunque numero tra 0 e 15 (ma si faccia attenzione a non usare 0, 1 e 2);

- "m":1: indica che il file risiedera' sul Microdrive 1;

- "numeri" e' il nome del file. Puo' essere composto al massimo da 10 caratteri.

Il comando OPEN esegue 2 funzioni importanti:

- crea un nuovo canale: "m";1;"numeri"

- assegna questo canale al flusso #4.

L'operazione dura circa 8 sec. perche' il calcolatore cerca prima se nella cartuccia esiste gia' un file di nome "numeri": e solo se non lo trova ne crea uno nuovo e si prepara a scriverci dei dati. Se il file esiste gia', viene aperto per la lettura. Se esiste un programma di nome "numeri", si ottiene il messaggio WRONG FILE TYPE (file di tipo errato).

INTRODUZIONE DEI DATI

Una volta aperto per la scrittura, un file e' pronto per ricevere dati. La scrittura dei dati su un file avviene con il comando PRINT seguito da # e dal numero di flusso a cui e' assegnato quel file. Per esempio per memorizzare i numeri da 1 a 10 e i loro quadrati nel file "numeri" si puo' usare il seguente programma:

```
10 OPEN #4;"m";1;"numeri"
20 FOR n=1 TO 10
30 PRINT #4,n'n*n
40 NEXT n
50 CLOSE #4
```

Viene spontaneo pensare che i dati vengano memorizzati nella cartuccia man mano che vengono stampati. In realta' il computer aspetta a scrivere sul nastro fino a quando la quantita' di questi dati non ammonta a 512 caratteri. In termini tecnici si dice che il Microdrive ha un BUFFER di 512 byte.

Per memorizzare COMUNQUE i dati in una cartuccia, anche se sono in quantita' inferiore a 512 caratteri, occorre usare il comando CLOSE che chiude il canale ed il file. Se questo comando non viene eseguito i dati rimasti nel buffer vanno persi.

CHIUSURA DI UN FILE

Chiudere un file fa si che il file sia memorizzato completamente nella cartuccia, chiude il canale (in questo caso "m";1;"numeri") e di conseguenza lascia il flusso (in questo caso #4) non assegnato ad alcun canale. Per chiudere un file e' necessario eseguire il comando CLOSE con il relativo numero di flusso. Nel nostro caso scrivete:

```
CLOSE #4
```

E' solo quando viene eseguito questo comando che lo schermo lampeggia per indicare che qualcosa viene memorizzato nella cartuccia. Si noti anche che, comè il comando OPEN, anche il comando CLOSE e' seguito automaticamente dal simbolo #.

Non e' possibile chiudere i flussi #0, #1, #2 o #3. Anche se vengono chiusi, i flussi #0 e #1 sono sempre assegnati al canale K; il flusso #2 al canale S, e il flusso #3 al canale P (vedi fine capitolo 5).

## LETTURA DEI DATI DA UN FILE

Per rileggere i dati dal file "numeri" si puo' usare questo programma:

```
10 OPEN #4;"m";1;"numeri"
20 FOR b=1 TO 10
30 INPUT #4;m;n
40 PRINT "Il quadrato di ";m;"e'";n
50 NEXT b
60 CLOSE #4
```

Il comando di apertura e' lo stesso usato per scrivere sul file, ma siccome il file "numeri" esiste gia', il canale "m";1;"numeri" e' aperto per la lettura, e non puo' quindi ricevere dati dal calcolatore. Qualunque tentativo di scrivere sul flusso #4 da' errore.

Per leggere un file si puo' anche usare la funzione INKEY$, che legge un carattere alla volta. Per rendervi conto di come funziona, provate questo programma:

```
10 OPEN #11;"m";1;"listato"
20 LIST #11
30 CLOSE #11
40 OPEN #12;"m";1;"listato"
50 PRINT INKEY$#12;
60 GO TO 50
```

Questo programma termina con il messaggio "END OF FILE" (fine del file), poiche' la funzione INKEY$ cerca di leggere anche un'altro carattere dopo aver letto l'ultimo. A causa di cio' il canale rimane aperto e il flusso #12 non puo' piu' essere usato. Eseguite quindi:

```
CLOSE #12
```

## OSSERVAZIONI SU PRINT E INPUT

Dato che i comandi PRINT e INPUT sono stati pensati per operare con il video e con la tastiera, occorre tenere conto delle caratteristiche di questi per adoperarli con i file.

## SEPARATORI

Il comando PRINT usa 3 diversi separatori:

- il ; (punto e virgola) che non stampa niente;

- la , (virgola) che stampa automaticamente tanti spazi quanti ne occorrono per arrivare all'inizio della mezza linea successiva;

- l ' (apostrofo) che causa lo spostamento a capo (ovvero inserisce un carattere ENTER).

Dal momento che il comando INPUT legge i dati introdotti soltanto quando viene premuto ENTER, anche quando legge dei dati da un file deve incontrare il carattere ENTER. Quindi per scrivere dei dati, che possano essere letti in variabili diverse da un comando INPUT occorre separarli con un carattere ENTER. Cio' si ottiene scrivendoli uno per uno:

```
PRINT#4;2
PRINT#4;3
```

oppure separandoli con l'apostrofo:

```
PRINT#4;2'3
```

Occorre anche prestare attenzione all'uso del comando INPUT. Come noto INPUT stampa nella parte bassa dello schermo tutto quello che puo' venire normalmente stampato da un comando PRINT. Ma, leggendo da un file, il file e' aperto solo per la lettura e non puo' ricevere dati. Se INPUT cerca di scrivere qualcosa si ha l'errore WRITING TO A "READ" FILE (tentativo di scrittura su un file aperto per lettura). In pratica le diverse variabili che devono essere lette da un file con un comando INPUT DEVONO ESSERE SEPARATE DA UN PUNTO E VIRGOLA. Per esempio:

```
INPUT #4;a;b
```

Occorre prestare particolare attenzione quando i dati contengono il carattere " (virgolette), infatti il comando INPUT considera il carattere " come indicatore di fine stringa. In questo caso occorre usare, come sempre:

```
INPUT #4;LINE a$
```

invece di:

```
INPUT #4;a$
```

CAMBIARE FLUSSO

Lo stesso comando PRINT puo' contenere informazioni per diversi flussi. Questo programma stampa "uno" sullo schermo, "due" sul file del Microdrive 1 di nome "cifra"; "tre" alla stazione 1 della rete (vedi capitolo 7) e "quattro" sulla linea successiva dello schermo.

```
10 OPEN #4;"m";1;"cifre"
20 OPEN #5;"n";1
30 PRINT "uno";#4;"due"'#5;"tre"'#2'"quattro"
40 CLOSE #4
50 CLOSE #5
```

AVVERTENZA SU PAPER E INK

Dopo aver usato un canale diverso dallo schermo, puo' succedere che i comandi PAPER e INK non funzionino piu'. In questo caso scrivete:

```
PRINT;
```

prima di usare PAPER e INK.

Se avete qualche dubbio sui comandi PRINT e INPUT, leggete il capitolo 24 del manuale in italiano di programmazione Basic.

IL CATALOGO DEI FILE

Tutti i file che vengono scritti su una cartuccia vengono di volta in volta inseriti nel catalogo. Quindi per vedere quale sia il contenuto di una cartuccia basta inserirla nel Microdrive e dare il comando di catalogo. Per esempio:

```
CAT 1
```

E lo schermo mostrera':

- il nome della cartucccia

- i nomi di tutti i file

- la quantita' di memoria disponibile nella cartuccia (in kilobyte).

La stampa del catalogo puo' anche essere eseguita su un flusso diverso dal video con la sintassi:

CAT # numero;numero

Il primo numero e' il numero di flusso e il secondo numero e' il numero del Microdrive. Questo serve per mandare il catalogo alla stampante o ad un file per essere poi letto da un programma.

## PROTEZIONE DI UN FILE

E' possibile impedire che il nome di un file venga stampato sul catalogo. Questa protezione si ottiene dando al file un nome che cominci col carattere il cui codice e' 0. Provate questo programma:

```
10 OPEN #4,"m";1;CHR$ 0+"risultati"
20 FOR n=1 TO 15
30 PRINT #4;n;n*n
40 NEXT n
50 CLOSE #4
```

e scrivete:

CAT 1

Noterete che il nome del file non appare nel catalogo. Fate quindi molta attenzione a non dimenticare mai i nomi dei file protetti e scriveteli da qualche parte non appena li create. Nessuno puo' sapere dove si trova e a che nome risponde un file protetto.

## COME ESTENDERE UN FILE

Si presenta spesso la necessita' di estendere un file. Per esempio si potrebbe desiderare di aggiungere al file "numeri" i quadrati dei numeri da 10 a 20. Siccome pero' un file non puo' essere riaperto per la scrittura, e' necessario:

- fare una nuova versione del file con un nome diverso;

- trasferire i dati del vecchio file alla nuova versione;

- aggiungere i nuovi dati;

- cancellare il vecchio file.

Provate facendo girare questo programma:

```
 10 OPEN #4;"m";1;"numeri":REM Per la lettura
 20 OPEN #5;"m";1;"numeri1":REM Per la scrittura
 30 FOR f=1 TO 10
 40 INPUT #4;m;n
 50 PRINT #5;m'n
 60 NEXT f
 70 FOR n=11 TO 20
 80 PRINT #5;n'n*n
 90 NEXT n
100 CLOSE #4: CLOSE #5
```

Ora controllate che sul Microdrive esistano due file, "numeri" e "numeri1", scrivete:

```
CAT 1
```

e cancellate il vecchio file con:

```
ERASE "m";1;"numeri"
```

e controllate che sia stato realmente cancellato con:

```
CAT 1.
```

Ora sul Microdrive esiste solo il file "numeri1" che contiene i numeri da 1 a 20 e i relativi quadrati.

-----------------

## 7. LA RETE LOCALE

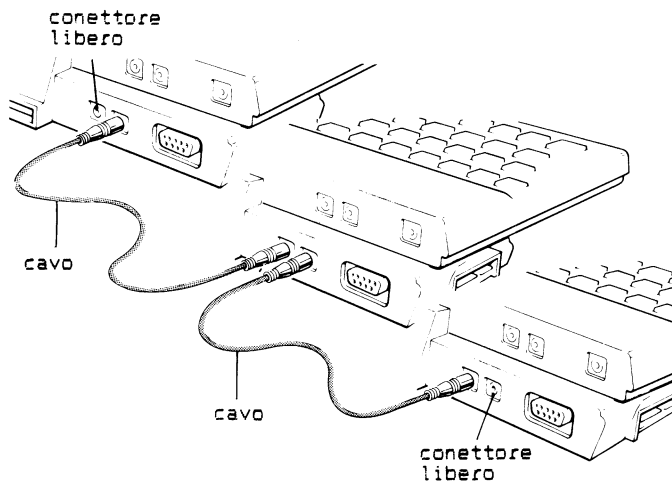
-----------------

### COME REALIZZARE UNA RETE

La rete locale, che chiameremo piu' semplicemente rete, permette a piu' Spectrum di scambiarsi programmi e dati e di fare giochi per piu' giocatori in cui ognuno e' davanti al suo terminale. Il vantaggio piu' immediato e' che uno solo degli utenti della rete deve battere o caricare il programma, cioe' molti Spectrum possono usufruire di un solo Microdrive.

La rete si dice locale perche' i calcolatori devono essere a distanza estremamente ravvicinata, al massimo nell'ordine di alcuni metri. Un collegamento a lunga distanza puo' essere realizzato su linea telefonica mediante l'impiego di modem.

La rete locale puo' comprendere da 2 a 64 Spectrum, ognuno dei quali deve essere equipaggiato con un interfaccia ZX 1. Le diverse interfacce devono essere collegate una all'altra come illustrato in figura.

Si faccia particolare attenzione per evitare di creare un anello di calcolatori: i computer alle due estremita' della rete non devono essere mai collegati tra di loro ma devono essere lasciati con una presa libera.

Lavorando con un calcolatore collegato ad una rete bisogna fare attenzione a non accenderlo o spegnerlo durante uno scambio di dati. La rete funziona anche se alcuni calcolatori della rete sono spenti e tutti i calcolatori possono venire accesi o spenti liberamente quando non c'e' nessuna comunicazione in corso.

Perche' la rete possa funzionare occorre che ogni terminale (calcolatore Spectrum) abbbia un diverso numero di identificazione. Occorre quindi stabilire tale numero e poi eseguire su ogni terminale il comando:

```
FORMAT "n";numero
```

dove numero e' il numero prescelto per la stazione su cui viene eseguito il comando.

Tuttavia se sulla rete ci sono soltanto due calcolatori, entrambi possono usare lo stesso numero di stazione. In questo caso non occorre usare il comando FORMAT, dato che ognuno viene definito implicitamente come stazione 1.

TRASMISSIONE DI PROGRAMMI LUNGO LA RETE

La procedura si spiega meglio con un esempio. Supponiamo che voi siate la stazione numero 1 e che un vostro amico sia la stazione numero 2. Se volete mandare al vostro amico questo programma:

```
10 REM radici
20 FOR n=1 TO 10
30 PRINT n,SQR n
40 NEXT n
```

Dopo aver battuto il programma occorre dare il comando:

```
SAVE *"n";2
```

(si noti che per le operazioni di trasferimento su rete non si usano i nomi dei programmi).

Perche' l'operazione sia portata a termine occorre che il vostro amico abbia dato:

```
FORMAT "n";2
```

seguito da:

```
LOAD *"n";1
```

Eseguita questa procedura il numero 2 ha caricato una copia 'del programma. Se provate noterete che, quando il computer sta aspettando di inviare o di caricare un programma sulla rete, il bordo dello schermo diventa nero fino a quando il programma non e' stato trasferito. Il vostro computer non manda il programma fino a quando quello del vostro amico non e' pronto a riceverlo, il suo computer dall'altra parte aspetta che voi lo mandiate. Provate a vedere cosa succede se viene inserito prima il comando SAVE o prima il comando LOAD.

Se la stazione 2 vuole verificare di avere una copia corretta del programma, deve scrivere:

```
VERIFY *"n";1
```

mentre voi dovete inviare un'altra volta il programma sulla rete ripetendo:

```
SAVE *"n";2
```

SAVE e' infatti l'unico modo di trasmettere un programma, mentre LOAD, VERIFY e MERGE sono tre modi diversi di riceverlo.

Il "gioco della rete" spiegato nell'appendice 1 e' un buon esempio di come giocare con 2 calcolatori.

COME INVIARE I DATI SULLA RETE

Supponete di voler inviare ad un vostro amico dei dati invece di un programma. Innanzi tutto dovete aprire un canale, per es. con:

```
OPEN #4;"n";2
```

che definisce la stazione 2 come canale sulla rete ("n"), e assegna tale canale al flusso #4. ora quando dei dati vengono inviati sul flusso #4, vengono inseriti sulla rete insieme a un dato che indica la stazione che li ha inviati.

Analogamente se scrivete INPUT #4;"n";2 il computer aspetta delle informazioni provenienti dal vostro amico.

Ora scrivete questo programma:

```
10 OPEN #4;"n";2: REM Per scrittura
20 INPUT a$: PRINT #4;a$
30 CLOSE #4
40 OPEN #4;"n";2: REM Per lettura
50 INPUT #4;b$: PRINT b$
60 CLOSE #4
70 GO TO 10
```

e poi scrivete:

```
SAVE *"n";2
```

come prima il vostro amico deve scrivere:

```
FORMAT "n";2
```

e poi:

```
LOAD *"n";1
```

Ora fate girare il vostro programma e dite al vostro amico di correggere le linee 10 e 40 in modo tale che si riferiscano alla stazione 1 e non alla stazione 2, e poi di eseguire:

```
GO TO 40
```

Ora siete pronti per iniziare una conversazione; confrontando il programma con quello che succede rendetevi conto che:

- tutto quello che scrivete sul flusso numero 4 e' bufferizzato. Questo significa che non viene inviato sulla rete fino a quando non ha raggiunto una certa quantita' di caratteri. Quindi ogni volta che terminate un messaggio dovete chiudere il flusso, la qual cosa produce la trasmissione dei dati anche se il buffer non e' pieno. Si ricordi che il buffer della rete e' di 255 caratteri;

- i vostri messaggi recano una indicazione sulla stazione che li ha trasmessi; quindi se il vostro amico sta aspettando un messaggio da un canale differente, il vostro messaggio viene ignorato. Se il vostro messaggio e' stato ignorato, non vedrete apparire sul video il messaggio OK e il bordo del vostro schermo rimarra' nero fino a quando il vostro amico eseguira' le operazioni corrette;

- anche il comando INKEY$ puo' essere usato per leggere la rete. A differenza del comando INPUT, che aspetta che qualcosa sia mandato, questo ritorna il primo carattere gia' trasmesso e contenuto nel buffer o in attesa di essere trasmesso. Se nessun dato e' in arrivo ritorna una stringa vuota. Questo modo di funzionamento e' chiamato polling. INKEY$ si comporta con la rete esattamente come con la tastiera.

Il seguente programma stampa qualunque cosa venga inviato dalla stazione 1:

```
10 OPEN #8;"n";1
20 PRINT INKEY$#8;
30 GO TO 20
```

Per maggiori informazioni su INKEY$ si consulti il capitolo 27 del manuale italiano di programmazione Basic.

TRASMISSIONE COLLETTIVA

La rete e' dotata di un particolare canale chiamato canale di trasmissione collettiva, che viene indicato da "n";0. Leggere da questo canale fa si che venga ricevuto ogni messaggio che qualunque stazione della rete vi trasmette. Trasmettere su questo canale fa si che il messaggio possa essere letto da chiunque sia in attesa sul canale "n";0.

Questa funzione puo' essere molto utile in una scuola, se ogni allievo ha uno Spectrum ma solo l'insegnante ha il Microdrive.

Per esempio se l'insegnante vuole trasmettere un programma a tutti, gli allievi devono scrivere per primi:

```
LOAD *"n";0
```

e mettersi in attesa del programma; poi l'insegnante deve scrivere:

```
SAVE *"n";0
```

Nella trasmissione collettiva, diversamente che con i messaggi individuali, il computer trasmittente non aspetta che la stazione ricevente sia pronta a ricevere, ma trasmette istantaneamente. In questo modo ovviamente chi invia il programma non ha nessun modo di sapere se e' stato ricevuto. Inoltre la funzione INKEY$ non esegue il polling sul canale di trasmissione collettiva, ma funziona come INPUT aspettando che qualcosa venga inviato.

Il canale 0 e' anche un modo di scambiare dati tra due Spectrum senza doversi preoccupare del numero di stazione, occorre pero' che il comando di ricezione (per es. LOAD) sia dato per primo.

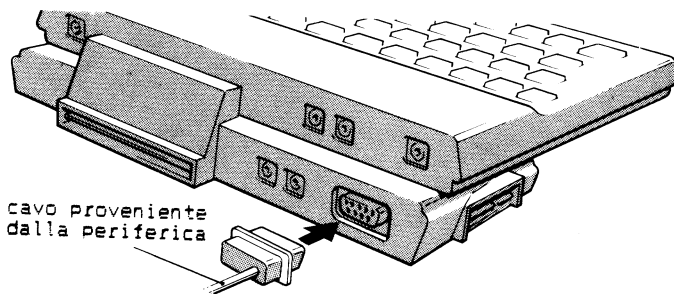
---

# 8. L'ÍNTERFACCIA RS232

---

## COME COLLEGARE LE PERIFERICHE ALL'INTERFACCIA RS232

Come si e' appreso dalla lettura del manuale Basic, il set dei caratteri dello Spectrum comprende sia simboli semplici, lettere, cifre, ecc., sia simboli composti, chiamati token, che sono le parole chiave, i nomi delle funzioni, ecc. Tutti questi sono caratteri di un solo byte e possono essere trasmessi e ricevuti dall'interfaccia RS232 e da tutte le periferiche seriali compatibili, per esempio una stampante, un modem, un'altra interfaccia RS232 collegata ad un altro calcolatore di qualunque tipo.

Per collegare una di queste periferiche all'interfaccia RS232, occorre usare un cavo con un connettore a vaschetta a 9 poli da un lato e l'opportuno connettore per la periferica dall'altro. Il connettore a vaschetta deve essere inserito nel retro dell'interfaccia ZX1 nella presa RS232 come mostrato nella figura. Per i dettagli di collegamento si consulti l'appendice 4.

Prima di poter usare l'interfaccia bisogna stabilire alcuni parametri della periferica. Tali parametri sono:

- la funzione di line feed automatico, che nella maggior parte dei casi deve essere disabilitata. Lo Spectrum scrive un ritorno di carrello e un avanzamento di linea sul canale "t", ma solo il ritorno di carrello su un canale "b". I canali "t" e "b" sono spiegati piu' avanti.

- la parita' on/off, che deve essere disabilitata.

- il numero dei bit, tenendo conto che lo Spectrum ne scrive 8.

- il numero dei bit di stop, tenendo conto che lo Spectrum usa un solo bit di stop.

- la baud rate, cioe' il numero di bit per sec. Lo Spectrum puo' lavorare a tutte le baud rate standard, cioe':
  50, 110, 300, 600, 1200, 2400, 4800, 9600 e 19200.

E' opportuno far usare alla periferica la baud rate piu' elevata possibile e in seguito adattare lo Spectrum a tale velocita'.

Come eseguire queste regolazioni e' spiegato nel manuale della periferica in questione.

## I CANALI "T" E "B"

L'interfaccia RS232 usa due diversi canali, il canale "t" e il canale "b".

## IL CANALE "T"

Il canale "t" o canale testo e' normalmente usato per mandare listati. Il canale "t" ha questo effetto sul set dei caratteri:

| | |
|---|---|
| 0-31 | (i codici di controllo) non sono inviati, salvo il 13 (ritorno di carrello = ENTER) che viene inviato come 13 seguito da 10 (ritorno di carrello e avanzamento di linea). |
| 32-127 | (i codici ASCII) sono trasmessi inalterati. |

128-164 (i caratteri grafici) non sono inviati. Al loro posto e' inviato il carattere "?" (codice 63).

165-255 (i tokens) sono trasformati in caratteri ASCII, codici 32-127.

Durante INPUT e INKEY$, il canale "t" aspetta un carattere a 7 bit e ignora di conseguenza l'ottavo bit, se c'e'.

Per usare il canale "t" occorre prima indicare quale baud rate si desidera usare. Cio' si ottiene con:

```
FORMAT "t"; baud rate
```

baud rate deve essere la stessa alla quale e' predisposta la periferica.

Ora per assegnare un flusso a questo canale ed inviarvi un listato occorre scrivere:

```
OPEN #3;"t"
LLIST
```

Il bordo dello schermo lampeggia e il listato viene mandato alla periferica. Si ricordi che LLIST e' un'abbreviazione di LIST #3. La trasmissione non avviene se non e' collegata alcuna periferica. Ora con:

```
LPRINT "questo e' un messaggio corto".
```

anche questo messaggio sara' mandato alla periferica. Ora chiudete il flusso.

Se la vostra interfaccia ZX1 e' collegata ad un terminale o ad una periferica RS232, capace di trasmettere dati, questi possono essere ricevuti. Provate questo programma:

```
10 FORMAT "t";600
20 OPEN #4;"t"
30 PRINT INKEY$#4;
40 GO TO 30
```

Eseguitelo e qualunque cosa trasmessa dal terminale o dalla periferica appare stampata sullo schermo. Naturalmente siate certi di usare la baud rate adatta.

## IL CANALE B

Il canale "b" (o canale binario) trasmette integralmente il codice a 8 bit usato dallo Spectrum e consente di mandare caratteri di controllo alle stampanti, ecc.

- Con INPUT e INKEY$ il canale "b" si aspetta un carattere di 8 bit.

- SAVE e LOAD funzionano solo sul canale b.

Se usate un modem per collegare il vostro calcolatore con un altro attraverso una linea telefonica o se vi collegate ad un altro tipo di calcolatore che ha un'interfaccia RS232, avete l'opportunita' di memorizzare e caricare programmi e dati attraverso l'interfaccia RS232. Per fare cio' occorre anzitutto stabilire la baud rate con:

```
FORMAT "b"; baud rate
```

baud rate, come al solito, deve essere la stessa della periferica.

Poi per esempio:

```
10 REM a caso
20 FOR n=1 TO 10
30 PRINT n,n*RND
40 NEXT n
```

seguito da:

```
SAVE *"b"
```

Se il programma viene inviato tramite un modem chi lo aspetta dall'altra parte della linea telefonica deve ovviamente scrivere:

```
LOAD *"b"
```

Come per tutte le operazioni di SAVE e di LOAD si possono usare le estensioni, per esempio:

```
SAVE *"b"; SCREEN$
```

oppure

```
SAVE *"b";LINE numero
```

COME INVIARE I CARATTERI DI CONTROLLO

Molte stampanti permettono funzioni speciali ottenibili tramite dei caratteri di controllo come, ad esempio, la stampa a larghezza doppia. Per mandare questi caratteri, occorre usare il canale b. Si ricordi che con il canale "b" il ritorno di carrello (=ENTER) viene inviato senza l'avanzamento di linea. Quindi occorre o cambiare le caratteristiche della stampante o inviare anche il line feed (codice ASCII 10).

Per esempio supponete che il codice di controllo per la stampa in larghezza doppia sia 14. Scrivete:

```
10 OPEN #4;"b"
20 PRINT #4;"LARGHEZZA NORMALE"
30 PRINT #4;CHR$ 14;"LARGHEZZA DOPPIA"
40 CLOSE #4
```

Ovviamente occorre verificare caso per caso quali siano i caratteri di controllo sul manuale della stampante.

E' possibile utilizzare contemporaneamente il canale "b" e il canale"t";provate questo programma:

```
10 OPEN #5;"b"
20 OPEN #6;"t"
30 PRINT #5;CHR$ 14;
40 LIST #6
50 CLOSE #5: CLOSE #6
```

che produce un listato in larghezza doppia, se la stampante lo consente e riconosce il carattere di codice 14 come attivatore.

------------------

# 9. IL COMANDO MOVE

------------------

Fino a questo momento abbiamo esaminato soltanto i trasferimenti di dati dai programmi ai canali e viceversa. Il comando MOVE invece permette di muovere i dati da un canale all'altro. Per esempio per muovere i dati dalla tastiera allo schermo, si puo' usare il comando:

```
MOVE #1 to #2
```

che produce la stampa sullo schermo di qualunque carattere battuto sulla tastiera anche se non viene visualizzato il cursore. Si faccia attenzione perche' il tasto BREAK non funziona ma produce soltanto la stampa di uno spazio. Per sbloccare il calcolatore occorre premere ENTER fino a quando la posizione di stampa non raggiunge la linea piu' bassa dello schermo e poi, quando il computer chiede SCROLL ?, premere BREAK. Si faccia attenzione a non usare questo comando per spostare i dati dalla tastiera ad un altro flusso perche' si potrebbe bloccare il sistema.

Il comando MOVE puo' anche essere utilizzato per esaminare i file memorizzati in una cartuccia. Per esempio, supposto che sia stato memorizzato il file "numeri" (vedi capitolo 6), si puo' esaminare il suo contenuto con:

```
MOVE "m";1;"numeri" to #2
```

Con il comando MOVE non occorre aprire o chiudere il file, perche' questa operazione viene eseguita automaticamente.
MOVE puo' anche essere usato per fare copie di file, per esempio:

```
MOVE "m";1;"numeri" TO "m";1;"numeri2"
```

apre un flusso per leggere il file "numeri" e un altro flusso per scrivere il nuovo file "numeri2"; quindi legge i dati da "numeri" e li scrive in "numeri2" e poi chiude entrambi i flussi.

Move accetta sia i numeri che specificano i flussi, (per esempio #4) sia gli identificatori dei canali (per

es. "m";1;"numeri"). Si noti comunque che i flussi predefiniti, cioe' quelli dal #0 al #3, non possono essere sostituiti dagli identificatori dei canali K, S o P, ai quali sono implicitamente assegnati.

MOVE puo' essere usato anche per trasferire dati da un Microdrive all'altro. Per esempio:

```
MOVE "m";1;"numeri" TO "m";2;"numeri2"
```

Si noti che MOVE funziona solo con i file di dati: per fare una copia di un programma bisogna caricarlo e poi rimemorizzarlo. E' comunque una buona regola fare sempre delle copie di qualunque insieme di dati o programma importante.

Il comando MOVE puo' essere anche usato per mandare i dati alla stampante. Se avete collegata una ZX Printer provate:

```
MOVE "m";1;"numeri" to #3
```

## IL PROGRAMMA DI SERVIZIO PER LA STAMPANTE (PRINTER SERVER)

Questo programma, contenuto nella cartuccia dimostrativa, permette ad uno Spectrum di una rete di controllare una stampante RS232. La stampante puo' essere usata da tutti gli altri calcolatori collegati alla rete, cosa particolarmente utile se un gruppo di persone che sta usando piu' Spectrum ha una stampante di alta qualita' che desidera dividere. Il programma evidenzia anche un potente uso del comando MOVE.

Il computer di servizio per la stampante deve essere sempre la stazione 64 e deve entrare in contatto con la stazione 62 (tale collegamento serve a stabilire il contatto). In questo modo chi invia i dati usa temporaneamente il numero di stazione 62 e manda il suo proprio numero di stazione, dal quale intende muovere un file verso il canale t.

Per fare diventare uno Spectrum una stazione di servizio per stampante si usa questo programma:

```
10 FORMAT "n";64
20 OPEN #4;"n";62: INPUT #4;a$: CLOSE #4
30 MOVE "n";CODE a$ TO "t"
40 OPEN #4;"b";62: PRINT #4;CHR$ 12: CLOSE #4: RUN
```

(la linea 40 manda un form feed)

L'utilizzatore invece deve usare il programma seguente. La stazione trasmittente si attribuisce temporaneamente il #62 e invia il suo proprio numero di stazione, quindi ritorna al suo vero numero e invia i dati che devono essere stampati alla linea 160 del programma (in questo caso il listato).

```
110 LET stazione=1: REM Proprio numero di stazione
120 FORMAT "n";62
130 OPEN #4;"n";64: PRINT #4;CHR$ stazione: CLOSE #4
140 FORMAT "n";stazione
150 OPEN #4;"n";64
160 LIST #4
170 CLOSE #4
```

"numero" alla linea 10 e' il numero proprio della stazione trasmittente.

---

APPENDICE 1.

IL GIOCO DELLA RETE (NET GAME)

---

Questo programma, contenuto nella cartuccia dimostrativa fornita col Microdrive, e' un gioco per due Spectrum collegati attraverso la rete locale. Alcune parti di questo programma possono tornare utili nei vostri programmi personali.

IL GIOCO

Per giocare, voi ed il vostro avversario dovete pensare un numero tra 1 e 100. Vince chi riesce ad indovinare per primo il numero pensato dall'altro. Ad ogni tentativo il vostro computer vi indica se siete piu' o meno vicini al numero da indovinare.

IL PROGRAMMA

La subroutine alla linea 500 decide chi e' l'utente #1 e chi e' l'utente #2; il #1 e' il primo a trasmettere. quando vi scambiate i tentativi uno di voi usa la subroutine alla linea 1100 e l'altro alla linea 1200.

Per decidere chi sia il numero 1, il vostro computer manda al computer del vostro avversario il messaggio "1" e attende un messaggio di risposta. Se riceve un altro "1" significa che il computer del vostro avversario e' stato acceso dopo il vostro e quindi il vostro computer manda un "2" al computer dell'avversario e si attribuisce il numero 1. Se invece il computer riceve un "2" significa che il computer del vostro avversario era gia' acceso quando il vostro computer ha inviato "1" e quindi il vostro computer e' il numero 2.

Se entrambi i computer iniziano il gioco allo stesso istante entrano in collisione e sara' necessario interrompere il programma e ricominciare da capo.

Il corpo principale del programma si occupa di scambiare i nomi, di leggere il numero segreto (che non viene trasmesso) e di confrontare i tentativi. Prima sono trasmessi i tentativi e poi le risposte.

La linea 190 segnala la vincita e offre la possibilita' di continuare il gioco.

```
   5 LET a$="n"
  10 GOSUB 500
  20 PRINT ;: BORDER 1: PAPER 1:
 INK 7: CLS
  30 PRINT "GIOCO INDOVINA IL NU
MERO"''"Introdurre il proprio nu
mero"'"segreto e cercare di indo
vinare"'"quello dell'avversario"
  35 IF a$="s" OR a$="S" THEN GO
 TO 50
  40 INPUT "Scrivi il tuo nome:
";n$
  50 LET a$=n$: PRINT ''"Salve "
;a$
  60 GO SUB 1000+100*user
  70 PRINT "Stai giocando contro
 ";b$
  75 PRINT 'a$,b$
  80 INPUT "Pensa un numero da 1
 a 100: ";a
  90 IF a<1 OR a>100 OR a<>INT a
 THEN GO TO 80
 130 INPUT " Cerca di indovinare:
";b
 135 IF b<1 OR b>100 THEN GO TO
130
 140 LET a$=STR$ b: GO SUB 1000+
100*user
 150 LET c=ABS (a-VAL b$)
 160 IF c=0 THEN LETa$="GIUSTO!
": GO TO 170
 161 IF c<4 THEN LET a$="Vicinis
simo...": GO TO 170
 162 IF c<10 THEN LET a$="Molto
vicino": GO TO 170
 163 IF c<20 THEN LET a$="Vicino
": GO TO 170
 164 IF c<40 THEN LET a$="A meta
' strada": GO TO 170
 165 IF c<60 THEN LET a$="Lontan
o": GO TO 170
 166 LET a$="Molto lontano"
 170 GO SUB 1000+100*user
 180 PRINT b$,a$
 190 IF c=0 OR b$="GIUSTO!" THEN
 GO TO 210
 200 GO TO 130
 210 IF b$="GIUSTO!" THEN PRINT
FLASH 1;"VITTORIA": FOR n=0 TO 7
: BORDER n: BEEP .1,n: BEEP .1,n
```

```
+16: NEXT n: GO TO 230
 220 PRINT "Battuto!": FOR n=7 T
O 0 STEP -1: BORDER n: BEEP .2,n
: NEXT n
 230 BORDER 1: INPUT "Vuoi gioca
re ancora? (s/n)";a$
 240 IF a$="s" OR a$="S" THEN GO
 TO 20
 499 STOP
 500 OPEN #4;"n";0
 510 PRINT #4;"1"
 520 CLOSE #4
 530 OPEN #4;"n";0
 540 INPUT #4;a$
 545 CLOSE #4
 550 LET a=4: IF a$="1" THEN OPE
N #a;"n";0: PAUSE 5: PRINT #4;"2
": LET user=1
 560 IF a$="2"THEN LET user=2
 570 CLOSE #4
 580 FORMAT "n";user: RETURN
1100 OPEN #4;"n";3-user
1110 PRINT #4;a$
1120 CLOSE #4
1130 OPEN #4;"n";3-user
1140 INPUT #4;b$
1150 CLOSE #4
1160 RETURN
1200 OPEN #4;"n";3-user
1210 INPUT #4;b$
1220 CLOSE #4
1230 OPEN #4;"n";3-user
1240 PRINT #4;a$
1250 CLOSE #4
1260 RETURN
```

--------------------

## APPENDICE 2

## VARIABILI DI SISTEMA

--------------------

Oltre alle variabili di sistema listate nel capitolo 34 del manuale italiano di programmazione Basic, il Microdrive, la rete di lavoro locale e l'interfaccia RS232 usano delle altre variabili di sistema listate in questa appendice.

La nota in prima colonna ha questo significato: se e':

X - le variabili non devono assolutamente essere alterate, perche' si potrebbe causare l'arresto del sistema.

N - Alterare la variabile non ha un effetto distrutttivo sul sistema

Il numero che segue la nota e' il numero di byte della variabile. Per le variabili a 2 byte il primo e' il meno significativo, diversamente da quanto si potrebbe immaginare. Quindi per alterare in "v" il valore della variabile a 2 byte di indirizzo n, usate:

```
POKE n,v-256* INT (v/256)
POKE n+1,INT (v/256)
```

e per ottenere il suo valore si usa l'espressione:

```
PEEK n+256* PEEK (n+1)
```

| Note | Ind. | Nome | Contenuti |
|---|---|---|---|
| X1 | 23734 | FLAGS 3 | Flag |
| X2 | 23735 | VECTOR | Indirizzi usati per estendere l'interprete Basic |
| X10 | 23737 | SBRT | Subroutine di "paging" della ROM |
| 2 | 23747 | BAUD | Numero due byte che determina la baud rate calcolato come segue: BAUD=(3500000/(26*baud rate))-2 questa variabile puo' essere usata per ottenere baud rate non standard |
| 1 | 23749 | NTSTAT | Proprio numero di stazione |

| | | | |
|---|---|---|---|
| 1 | 23750 | IOBORD | Colore del bordo dello schermo usato durante l'input/output. E' possibile usare qualsiasi colore. |
| N2 | 23751 | SER-FL | Area di lavoro a due byte per RS232 |
| N2 | 23753 | SECTOR | Area di lavoro a due byte usata per il Microdrive |
| N2 | 23755 | CHADD- | Memorizzazione temporanea di CH-ADD |
| 1 | 23757 | NTRESP | Spazio per il codice di risposta della rete |
| 1 | 23758 | NTDEST | Inizio del buffer della rete. Contiene il numero della stazione di destinazione tra 0 e 64 |
| 1 | 23759 | NTSRCE | Numero della stazione trasmittente |
| X2 | 23760 | NTNUMB | Numero di blocco per la rete 0-65535 |
| N1 | 23762 | NTTYPE | Codice del tipo di intestazione |
| X1 | 23763 | NTLEN | Lunghezza del blocco di dati 0-255 |
| N1 | 23764 | NTDCS | Byte di controllo (checksum) del blocco di dati |
| N1 | 23765 | NTHCS | Byte di checksum del blocco di intestazione |
| N2 | 23766 | D-STR1 | Inizio dello specificatore di file a 8 byte. Contiene il numero a 2 byte del drive nell'intervallo 1-8 |
| N1 | 23768 | S-STR1 | Numero di flusso tra 1 e 15 |
| N1 | 23769 | L-STR1 | Tipo di device..."m", "n", "t" o "b" |
| N2 | 23770 | N-STR1 | Lunghezza del nome del file |
| N2 | 23772 | | Inizio del nome del file |
| N8 | 23774 | D-STR2 | Specificatore di file a 8 byte usato dai comandi MOVE e LOAD |
| N1 | 23782 | HD-00 | Inizio dell'area di lavoro per SAVE, LOAD, VERIFY e MERGE |

| | | | |
|---|---|---|---|
| N2 | 23783 | HD-0B | Lunghezza dei dati 0-65535 |
| N2 | 23785 | HD-0D | Inizio dei dati 0-65535 |
| N2 | 23787 | HD-0F | Lunghezza del programma 0-65535 |
| N2 | 23789 | HD-11 | Numero di linea |
| 1 | 23791 | COPIES | Numero di copie fatte da SAVE |
| | 23792 | | Inizio della MAPPA del Microdrive o dell'area delle informazioni di canale (CHANS). |

AVVERTENZE

1. L'apertura di un flusso verso il Microdrive o verso la rete richiede una certa quantita' di memoria libera con la quale creare il canale. Un canale di Microdrive e' lungo 595 byte, un canale della rete 276 byte. Questi canali sono creati dai comandi OPEN#, MOVE o da SAVE/LOAD/VERIFY/MERGE. Questo implica che qualunque programma che non lascia abbastanza spazio prima di RAMTOP produce il messaggio OUT OF MEMORY se si cerca di eseguire una di queste operazioni.

2. Un'altro effetto della creazione di questi buffer e' di spostare il linguaggio macchina memorizzato in un comando REM. Questo puo' creare problemi, percio' fate attenzione a locare i codici oggetto sempre sopra RAMTOP.

3. E' sconsigliabile usare BREAK durante un'operazione di scrittura sul Microdrive, cioe' quando il bordo dello schermo lampeggia, dato che si puo' lasciare un file non chiuso. ERASE cancella anche i file non chiusi ma impiega circa 30 sec. perche' esamina la cartuccia diverse volte alla ricerca dell'inesistente fine del file.

---

# APPENDICE 3

## CANALE DEI MICRODRIVE

---

Ogni volta che viene aperto un file, viene creata un'area chiamata CANALE nell'area di memoria etichettata "informazioni di canale" (CHANS). Quest'area viene generalmente puntata dal registro IX del software. Il canale ha una lunghezza di 595 byte e contiene il buffer di 512 byte.

I contenuti del canale sono i seguenti:

| | | |
|---|---|---|
| 0 | | Indirizzo 8 |
| 2 | | Indirizzo 8 |
| 4 | | "M" |
| 5 | | Indirizzo della subroutine di output nella ROM |
| 7 | | Indirizzo della routine di input nella ROM |
| 9 | | Indirizzo 595 |
| 11 | CHBYTE | Contatore del byte corrente che indica il prossimo byte che deve essere aggiunto o tolto dall'area dei dati. Tra 0 e 512 compresi. |
| 13 | CHREC | Numero di record. Indica la posizione del record nel file nell'intervallo 0-255 |
| 14 | CHNAME | Nome del file a 10 byte con spazi di riempimento |
| 24 | CHFLAG | Byte dei flag:<br>bit 0: settato....aperto per la scrittura.<br>resettato..aperto per la lettura.<br>bit 1-7....non utilizzati |
| 25 | CHDRIV | Numero del drive 0-7 |
| 26 | CHMAP | Indirizzo della mappa del Microdrive |

| | | |
|---|---|---|
| 28 | CHMAP | Primi 12 byte dell'intestazione: inizio dell'area di lavoro dell'intestazione |
| 40 | HDFLAG | Byte di flag; bit 0.... settato<br>bit 1-7...inutilizzati |
| 41 | HDNUMB | Numero di settore nell'intervallo 0-255 inutilizzato |
| 44 | HDNAME | Nome della cartuccia con spazi di riempimento |
| 54 | HDCHK | Byte di checksum sull'intestazione |
| 55 | | Primi 12 byte del blocco di dati: inizio dell'area di lavoro dei dati |
| 67 | RECFLG | Byte di flag; bit 0=0<br>bit 2=0 se file di scrittura<br>bit 2=1 se file di lettura<br>bit 3-7 non utilizzati |
| 68 | RECNUM | Numero di questo record nell'intervallo 0-255 |
| 69 | RECLEN | Numero di byte di dati di questo record 0-512 |
| 71 | RECNAM | Nome del file con spazi di riempimento |
| 81 | DESCHK | Checksum degli ultimi 14 byte |
| 82 | CHDATA | Inizio del buffer di 512 byte |
| 594 | DCHK | Checksum degli ultimi 512 byte |

MAPPA

Per ogni Microdrive con almeno un file aperto viene creata una MAPPA nell'area di memoria chiamata "Mappa del Microdrive". La mappa contiene 32 byte. Ogni bit corrisponde ad un settore del corrispondente Microdrive. Se tale settore contiene i dati o e' inutilizzabile, allora tale bit e' settato (=1). I bit sono numerati come segue: bit 0 byte 0= settore 0; bit 1 byte 0= settore 1; bit 0 byte 1= settore 8; e cosi' via.

CANALI DELLA RETE

Quando un flusso viene aperto verso la rete, viene creato un canale nell'area di memoria chiamata "informazioni di canale" (CHANS). Quest'area viene solitamente indirizzata dal registro IX del software. Il canale ha la lunghezza di 276 byte e contiene il buffer di 255 byte. Contenuti nel canale sono i seguenti:

| | | |
|---|---|---|
| 0 | | Indirizzo 8 |
| 2 | | Indirizzo 8 |
| 4 | | "N" |
| 5 | | Indirizzo nella ROM della subroutine di output |
| 7 | | Indirizzo nella ROM della subroutine di input |
| 9 | | Indirizzo 276 |
| 11 | NCIRIS | Numero della stazione destinazione |
| 12 | NCSELF | Numero di stazione di questo Spectrum |
| 13 | NCNUMB | Numero di blocco |
| 15 | NCTYPE | Codice che indica il tipo di blocco: 0=blocco dati, 1=blocco di fine file (EOF) |
| 16 | NCOBL | Numero di byte nel blocco di dati |
| 17 | NCDCS | Checksum dei dati |
| 18 | NCHCS | Checksum dell'intestazione |
| 19 | NCCUR | Posizione dell'ultimo carattere prelevato dal buffer |
| 20 | NCIBL | Numero di byte contenuti nel buffer di input |
| 21 | NCB | Inizio del buffer di 255 byte |

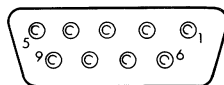
-----------------------------------

## APPENDICE 4

## COLLEGAMENTO DELL'INTERFACCIA RS232

-----------------------------------

Lo zoccolo dell'interfaccia RS232 dello Spectrum e' collegato come segue:

1. Non collegato
2. TX data (input)
3. RX data (output)
4. DTR (input) deve essere alto quando e' "ready"
5. CTS (output) deve essere alto quando e' "ready"
6. Non collegato
7. Massa (pull down)
8. Non collegato
9. + 9 volt (pull up)

Guardando il connettore sul retro dell'interfaccia ZX 1: La fila di contatti in alto e' numerata da 1 a 5 DA DESTRA VERSO SINISTRA, la fila in basso da 6 a 9 sempre DA DESTRA VERSO SINISTRA.

Un cavo RS232 che collega il connettore a vaschetta a 9 poli dell'interfaccia con il connettore standard a vaschetta a 25 poli delle periferiche puo' essere acquistato nei negozi specializzati e dalla SINCLAIR RESEARCH. Dalla parte del connettore a 25 poli il cavo e' collegato come segue:

2. TX data ................................... rosso
3. RX data ................................... bianco
5. CTS ....................................... blu
6. + 9 volt (normalmente DSR) ................ verde
7. massa ..................................... nero
20. DTR ...................................... giallo

------------

## APPENDICE 5

## MESSAGGI

------------

Il calcolatore collegato con l'interfaccia ZX 1 puo' produrre dei messaggi di errore che NON sono descritti nell'appendice B del manuale di programmazione Basic. Tali messaggi si riferiscono a situazioni in cui viene impiegata l'interfaccia e sono sempre seguiti dal numero di linea e dal numero di comando in cui si e' verificato l'errore. Ecco la lista dei messaggi:

CODE ERROR (ERRORE CON DATI BINARI)
Si e' cercato di caricare un file di tipo byte piu' grande dell'area di destinazione specificata nel comando LOAD.

DRIVE 'WRITE' PROTECTED (MICRODRIVE PROTETTO CONTRO LA SCRITTURA)
Si e' cercato di scrivere su una cartuccia a cui e' stata tolta la linguetta di protezione.

END OF FILE (FINE DEL FILE)
Questo errore e' gia' spiegato nel manuale Basic. Con l'interfaccia ZX 1 si presenta anche quando si cerca di leggere oltre la fine del file dal Microdrive, dalla rete o dall'interfaccia RS232. Un flusso che produce questo errore deve essere chiuso.

FILE NOT FOUND (FILE INESISTENTE)
Puo' significare sia che si e' cercato di caricare un file che non esiste, sia che parte del file non riesce ad essere riletto. Puo' succedere se il file non e' stato chiuso o se il file e' stato danneggiato, magari perche' la cartuccia e' stata lasciata inserita nel Microdrive durante l'accensione o lo spegnimento.

INVALID DEVICE EXPRESSION (IDENTIFICATORE DI PERIFERICA NON VALIDO)
E' stata specificata una periferica diversa da S,P,K,M,N,T o B. Questo errore si ha anche quando viene usato il punto e virgola invece della virgola coi canali S,P o K.

INVALID DRIVE NUMBER (NUMERO DI DRIVE NON VALIDO)
E' stato specificato un numero di Microdrive fuori

dall'intervallo 1-8.

INVALID NAME (NOME NON VALIDO)
E' stato specificato un nome di file con nessun carattere oppure con piu' di 10 caratteri.

INVALID STATION NUMBER (NUMERO DI STAZIONE NON VALIDO)
Si e' usato un numero di stazione fuori dall'intervallo 0-64 (per il comando FORMAT fuori dall'intervallo 1-64).

INVALID STREAM NUMBER (NUMERO DI FLUSSO NON VALIDO)
E' stato specificato un flusso con un numero non compreso nell'intervallo 0-15.

MERGE ERROR (ERRORE DI MERGE)
Si e' cercato di usare il comando MERGE con un file di dati o di byte. MERGE puo' essere usato solo con i programmi. Questo messaggio si ha anche quando si cerca di usare MERGE con un programma memorizzato con l'opzione LINE, questo a scopo di protezione.

MICRODRIVE FULL (MICRODRIVE PIENO)
Si e' cercato di memorizzare dati su una cartuccia che non ha abbastanza spazio libero per il file che si vuole memorizzare. Occorre ripetere l'operazione su una cartuccia che abbia abbastanza spazio libero. Se non si ha a disposizione un'altra cartuccia occorre cancellare qualche vecchio file da quella che si vuole usare. Il file aperto per la scrittura che non ha potuto essere memorizzato completamente non e' stato chiuso e deve essere cancellato; ci vorranno circa 30 secondi. RICORDATEVI DI CANCELLARE IL FILE CHE STAVATE CERCANDO DI MEMORIZZARE, SE APPARE NEL CATALOGO.

MICRODRIVE NOT PRESENT (MICRODRIVE NON PRONTO)
Si e' cercato di usare un Microdrive che non e' collegato all'interfaccia, un Microdrive che non contiene una cartuccia, o un Microdrive che contiene una cartuccia non formattata.

MISSING BAUD RATE (MANCA LA BAUD RATE)
La baud rate non e' stata specificata.

MISSING DRIVE NUMBER (MANCA IL NUMERO DEL DRIVE)
Il numero del drive non e' stato specificato.

MISSING NAME (MANCA UN NOME DI FILE)
Il nome di file non e' stato specificato.

MISSING STATION NUMBER (MANCA IL NUMERO DI STAZIONE)
Il numero di stazione della rete non e' stato specificato.

PROGRAM FINISHED (OLTREPASSATA LA FINE DEL PROGRAMMA)
Si e' cercato di eseguire una linea al di la' dell'ultima linea del programma. Questo messaggio si ha quando si usa un'istruzione GOTO seguita da un numero superiore al numero dell'ultima linea del programma o se viene usato il comando RUN senza un programma.

READING A 'WRITE' FILE (TENTATIVO DI LETTURA SU UN FILE DI SCRITTURA)
Si e' cercato di leggere dati da un file che non e' stato ancora chiuso e quindi e' ancora aperto per la scrittura.

STREAM ALREADY OPEN (FLUSSO GIA' APERTO)
Si e' cercato di usare OPEN con un flusso che e' gia' stato aperto verso un nuovo canale (M,N,T o B). Il flusso puo' essere aperto solo se prima viene chiuso.

VERIFICATION HAS FAILED (VERIFICA NEGATIVA)
E' stata rilevata una differenza tra il file specificato in un comando di VERIFY e quello contenuto in memoria. SIGNIFICA CHE E' NECESSARIO RIPETERE L'OPERAZIONE DI SAVE.

WRITING TO A 'READ' FILE (TENTATIVO DI SCRITTURA SU UN FILE DI LETTURA)
Si e' cercato di scrivere dati verso un file che esiste gia'. Il file esistente deve essere cancellato se non e' necessario, altrimenti deve essere usato un nome di file diverso. Questo messaggio si ha anche quando vengono usati separatori diversi dal punto e virgola (solitamente la virgola) in un comando di INPUT da file.

WRONG FILE TYPE (TIPO DI FILE SBAGLIATO)
Si e' cercato di usare INPUT o MOVE per un file creato con SAVE o di usare LOAD, VERIFY o MERGE con un file creato con OPEN e PRINT o di caricare una matrice o un file binario (CODE) come programma o viceversa. Invece di INPUT occorre usare LOAD, con LOAD occorre usare o le opzioni CODE e DATA oppure usare INPUT.

---------------

## APPENDICE 6

## IL BASIC ESTESO

---------------

L'interfaccia ZX1 estende il Basic residente dello Spectrum. Questa appendice tratta le estensioni e le modifiche.

### FLUSSI

I flussi sono specificati come #n, dove n e' un numero tra 1 e 15. I flussi 0, 1, 2 e 3 sono usati correntemente dal Basic. Il carattere # (obbligatorio) e' parte della parola chiave per i comandi OPEN# e CLOSE#.

### CANALI

Nel Basic esteso ci sono 7 tipi di canali; la tastiera (k=keyboard), lo schermo (s=screen), la ZX printer (p), l'interfaccia RS232 con formato testo (t), l'interfaccia RS232 con formato binario (b), la rete (n=network) e il Microdrive (m).

Ogni tipo di canale e' identificato dalla lettera; puo' essere data sia in maiuscolo che in minuscolo. La rete e il Microdrive richiedono delle informazioni supplementari per identificare completamente il canale.

Un canale della rete richiede un numero di stazione e viene specificato come "n";x dove x e' il numero di stazione nell'intervallo 0-64.

Un canale di Microdrive richiede un numero di Microdrive e un nome di file e viene specificato come "m";y;"nome" dove y e' il numero del Microdrive nell'intervallo 1-8 e "nome" e' una stringa lunga da 1 a 10 caratteri (non sono ammesse le stringhe nulle).

CAT — Produce la lista di tutti i file contenuti nella cartuccia inserita nel Microdrive y. La lista e' in ordine alfabetico preceduta dal nome della cartuccia e seguita dalla quantita' di memoria libera espressa in kilobyte.

| | |
|---|---|
| CAT #z;y | Invia il catalogo della cartuccia nel Microdrive y, col formato gia' descritto, al flusso #z. |
| CLOSE # flusso | Scollega un canale dal flusso specificato. Se il buffer contiene ancora qualche dato, questo viene o trasmesso sulla rete o memorizzato sul Microdrive, o perso se doveva essere letto. |
| ERASE "m";y;"nome" | Cancella il file specificato dalla cartuccia del Microdrive y (si noti che differentemente da LOAD,SAVE e MERGE non richiede l'asterisco). |
| FORMAT "m";y;"nome" | Prepara per l'uso la cartuccia nuova inserita nel Microdrive y. Il nome viene assegnato alla cartuccia e compare nel catalogo. |
| FORMAT "n";x | Attribuisce alla stazione su cui viene eseguito il numero x. |
| FORMAT "t";x<br>FORMAT "b";x | Seleziona la baud rate per l'interfaccia RS232 al valore x, che deve essere uno degli standard (50, 110, 300, 600, 1200, 2400, 4800, 9600, 19200). Baud rate non standard possono essere ottenute come indicato nella lista delle variabili di sistema (BAUD). |
| INKEY$ # flusso | Ritorna una stringa di lunghezza unitaria se e' disponibile un carattere sul flusso specificato e una stringa nulla ("") se non e' disponibile alcun carattere. Quest'istruzione ha significato solo se il flusso e' collegato alla rete o all'interfaccia RS232. |

| | |
|---|---|
| INPUT # flusso;variabile | Legge la variabile dal flusso specificato. Il flusso deve essere stato aperto in precedenza come canale di INPUT. E' importante notare che ogni elemento di stampa in un comando di INPUT viene scritto sul flusso, come avviene normalmente con la tastiera. Si ricordi che anche il separatore "virgola" scrive un carattere, per cui di solito non e' ammesso. L'opzione LINE funziona al solito modo. |
| LOAD * canale opzioni | Carica il programma, i dati o i codici dal canale specificato. Puo' essere usato solo dai canali "b", "n" e "m". Con LOAD* possono essere usate tutte le opzioni di LOAD (CODE, SCREEN$...). |
| MERGE * canale opzioni | Come LOAD ma non cancella le vecchie linee di programma o variabili a meno che non vengano sostituite dalle nuove con lo stesso numero di linea o nome. |
| MOVE sorgente TO destinazione | Trasferisce i dati dalla sorgente alla destinazione. La sorgente e la destinazione possono essere sia numeri di flussi che canali. L'esecuzione del comando ha termine soltanto quando viene incontrato un indicatore di fine file nella sorgente; questo si verifica solo se la sorgente e' la rete, un canale di Microdrive o un flusso collegato a uno di questi. Se la sorgente o la destinazione e' un canale, viene aperto prima dell'inizio del comando e automaticamente chiuso terminato il trasferimento. |

| | |
|---|---|
| OPEN # flusso, canale | Assegna il canale specificato al flusso specificato per permettere al Basic di leggere o scrivere dati verso quel canale. Il flusso deve essere stato chiuso in precedenza, ovvero non deve essere assegnato a niente salvo che a K,S o P (flussi #0-#3). |
| PRINT # flusso ... | Invia la sequenza di stampa (...) al flusso specificato. Il flusso deve essere stato aperto in precedenza come il canale di OUTPUT. La sequenza di stampa ha la sintassi usuale e puo' contenere altri # per cambiare il flusso di destinazione. |
| SAVE * canale opzioni | Memorizza i programmi, i dati o i codici binari al canale specificato. Si puo' usare solo con i canali "b", "n" o "m". Con SAVE * si possono usare tutte le opzioni di SAVE. |
| VERIFY * canali opzioni | Come LOAD ma i dati non vengono caricati bensi' confrontati con i dati contenuti in memoria. |

## INDICE ANALITICO

P



R



S



T



V



Z